Mechanical Engineering Publications

Mechanical Engineering

4-28-2014

Immersive Computing Technology to Investigate Tradeoffs Under Uncertainty in Disassembly Sequence Planning

Sara Behdad
University at Buffalo

Leif P. Berg
Iowa State University, lpberg@iastate.edu

Judy M. Vance
Iowa State University, jmvance@iastate.edu

Deborah Thurston
University of Illinois at Urbana-Champaign

Immersive Computing Technology to Investigate Tradeoffs Under Uncertainty in Disassembly Sequence Planning

Sara Behdad
Department of Mechanical and Aerospace Engineering,
Department of Industrial and Systems Engineering,
University at Buffalo, SUNY
Buffalo, NY 14260
e-mail: behdad1@illinois.edu

Leif Berg
Department of Mechanical Engineering,
Iowa State University,
2274 Howe Hall,
Ames, IA 50011
e-mail: lpberg@iastate.edu

Judy Vance
Department of Mechanical Engineering,
Iowa State University,
2274 Howe Hall,
Ames, IA 50011
e-mail: jmvance@iastate.edu

Deborah Thurston[1]
Department of Industrial and Enterprise Systems Engineering,
University of Illinois at Urbana-Champaign,
104 S. Mathews,
Urbana, IL 61801
e-mail: thurston@illinois.edu

The scientific and industrial communities have begun investigating the possibility of making product recovery economically viable. Disassembly sequence planning may be used to make end-of-life product take-back processes more cost effective. Much of the research involving disassembly sequence planning relies on mathematical optimization models. These models often require input data that is unavailable or can only be approximated with high uncertainty. In addition, there are few mathematical models that include consideration of the potential of product damage during disassembly operations. The emergence of Immersive Computing Technologies (ICT) enables designers to evaluate products without the need for physical prototypes. Utilizing unique 3D user interfaces, designers can investigate a multitude of potential disassembly operations without resorting to disassembly of actual products. The information obtained through immersive simulation can be used to determine the optimum disassembly sequence. The aim of this work is to apply a decision analytical approach in combination with immersive computing technology to optimize the disassembly sequence while considering trade-offs between two conflicting attributes: disassembly cost and damage estimation during disassembly operations. A wooden Burr puzzle is used as an example product test case. Immersive human computer interaction is used to determine input values for key variables in the mathematical model. The results demonstrate that the use of dynamic programming algorithms coupled with virtual disassembly simulation is an effective method for evaluating multiple attributes in disassembly sequence planning. This paper presents a decision analytical approach, combined with immersive computing techniques, to optimize the disassembly sequence. Future work will concentrate on creating better methods of estimating damage in virtual disassembly environments and using the immersive technology to further explore the feasible design space. [DOI: 10.1115/1.4025021]

1 Introduction

Among the various areas that affect the efficiency of end-of-life (EOL) product recovery operations, disassembly has been the focus of a large number of research projects [1]. In addition to EOL considerations, maintenance operations during the customer use phase of the product life cycle also often require disassembly. The efficiency of the disassembly sequence thus influences the profitability of both salvaging and maintenance activities. Exploring potential disassembly sequences early in the design process provides the opportunity to evaluate and perhaps modify the product design in ways that could improve the disassembly process.

Disassembly sequences are listings of subsequent disassembly actions conducted for separation of an assembly to its subassemblies [2]. Disassembly sequence planning may be conducted for a variety of objectives. Such objectives include the reusability of certain components, the recovery of components which still have embedded value, the removal of defective parts in the course of maintenance, assembly planning, etc. [3]. A good disassembly plan incorporates considerations for minimum disassembly time, low cost, minimum damage to components, operator safety, and ergonomics.

There are situations in which disassembly planning cannot be completed using physical prototypes, such as remote maintenance and repair in inaccessible or hazardous environments. While various algorithmic and optimization approaches have been developed to tackle the disassembly sequence planning problem, providing the input data for these approaches during the early design stage, or in cases in which physical prototypes are not available, is a challenge. In these situations, ICT can be employed to facilitate physical prototype simulations.

ICT places the user into a simulated 3D computer generated world. Through the use of stereo viewing, 3D position tracking and haptic (force feedback) devices, ICT allows users to interact with computer generated images/products using natural human motions. In this manner, users can manipulate digital representations of products in ways similar to how they would manipulate physical prototypes. ICT supports an ego-centric approach and manipulation of objects in real scale that is not possible using traditional computer interaction tools such as the monitor, mouse and keyboard. Kinesthetic feedback involved in self-awareness of body motions and spatial relationships is an important aspect in evaluating disassembly operations.

The aim of this research is to explore the coupling of a decision analytical approach and ICT to optimize a disassembly plan for reuse and recovery while considering trade-offs between two attributes: disassembly cost and the probability of damage. The proposed method models the decision maker's preference toward risk and allows the consideration of uncertainties in the disassembly process.

[1]Corresponding author.

Contributed by the Design Automation Committee of ASME for publication in the JOURNAL OF MECHANICAL DESIGN. Manuscript received January 15, 2012; final manuscript received June 27, 2013; published online April 28, 2014. Assoc. Editor: Karthik Ramani.

The remainder of this paper is organized as follows: Sect. 2 provides a brief review of related literature. Section 3 introduces the formulation of the disassembly sequence model, including a dynamic programming model incorporating utility theory to solve a multiattribute disassembly sequence planning problem. An example problem is introduced in Sec. 4 where ICT is used to determine the input parameters for the optimization. Section 5 presents results, and Sec. 6 presents overall conclusions and recommendations for future work.

2 Literature Review

The work presented here draws upon research in two distinct fields: multi-objective disassembly planning and disassembly/assembly using ICT.

Researchers have proposed various approaches to achieve disassembly sequence planning, including the disassembly tree approach, the disassembly Petri net, and the AND/OR graph based approach [4]. Although the primary objective of much of the research is to minimize disassembly cost, some research methods include other objectives as well. Hula et al. [5] developed a decision making methodology that determines how to maximize the environmental benefits of EOL operations while minimizing costs. McGovern and Gupta [6] applied an ant colony optimization metaheuristic for obtaining optimal or near-optimal solutions to the disassembly line balancing problem. They considered multiple objectives including minimizing the number of workstations, minimizing idle time, and balancing the line. Lee et al. [7] determined the disassembly schedules for end-of-life products subject to capacity restrictions. Some research has focused on selective disassembly for the purpose of maintenance in which the final status of the product or the target component is defined a priori [8]. The primary objective of much of the research has been to maximize the economic returns, or to maximize efficiency with respect to disassembly time and the number of removed components [5].

When the goal of disassembly is reuse or material recovery, additional considerations are warranted. An evaluation of the potential for reuse of various subassemblies will affect decisions regarding the best process plan. Estimates of material recovery will also influence the final plan. Some research integrates disassembly cost and the resulting cost of component EOL options together to find the optimal disassembly sequence [9–11].

Another consideration is that disassembly is primarily accomplished through human labor instead of the use of automated robotic assembly lines. One factor that has not received much attention is how to estimate the amount of damage that may occur during disassembly operations. Lambert [2] emphasizes the importance of considering potential product damage that may occur during the disassembly process resulting from human actions. Behdad and Thurston [12] developed a decision analytical approach to account for the uncertainties associated with the disassembly process, including damage estimates. In their model they employed mixed integer linear programming to find the optimal disassembly sequence, considering both cost and damage. The probability of damage was estimated using historical data gathered from previous disassembly operations. In situations where no historical data exists, another method is needed to generate the input data.

The literature described above shows that although disassembly is a complex and costly process, mathematical models are being developed with the goal of making that process more efficient. However, these models are themselves complex and require large amounts of data that can be very difficult to gather. Furthermore, even after the data is gathered, unavoidable uncertainty remains, due to the very nature of the disassembly process. This requires designers to consider the effect of unavoidable uncertainty due to variability in product condition, operator skill, etc.

ICT provides designers with new opportunities to gather this difficult to obtain data in a way that includes the effect of the human operator. The proliferation of ICT has enabled engineers to attack real world problems in industry [13]. Several researchers have been exploring how this technology might improve assembly training and planning; however, few have examined these techniques for disassembly [14]. Jayaram et al. [15] developed a general purpose ICT application called VADE, that allows users to simulate assembly operations and factory and facility layouts. Seth et al. [16] developed SHARP, that supported two-handed interaction with haptic (force feedback) to simulate manual assembly operations.

In order to simulate realistic virtual object interaction, a Virtual Reality (VR) assembly application must provide a method for detecting object collisions and generating interaction forces. A significant challenge is the need to compute collisions and forces over very short time frames (60–1000 Hz) to support interactive manipulation of complex CAD models. Lin and Gottaschalk [17] and Jimenez et al. [18] present a survey of 3D collision detection algorithms and Borro et al. [19] organize these algorithms into a taxonomy. Voxel-based methods, such as Voxmap pointshell, have proven especially effective in simulating full 6DOF haptic interactions [20], but the reliance on using approximate geometry for collision detection presents a challenge when faced with assembly of low clearance parts. Faas and Vance [21] present a method of pointshell shrinking to support low clearance virtual assembly tasks, and Seth et al. [22] developed a tiered approach using both exact and approximate geometry to support low clearance assembly.

Researchers have proposed various approaches to disassembly sequence planning. Dong and Arndt [23] present a comprehensive overview of disassembly sequence planning including some methods based on ICT. Ritchie et al. [24] first proposed combining knowledge capture and ICT to support assembly methods planning. An application by Dewar et al. [25] logged user interactions and created assembly plans from logs. Bullinger et al. [26] presented an application that generates a precedence graph based on user interaction. Time and cost of disassembly were calculated. Aleotti and Caselli [27] applied the concept of physics-based modeling in virtual reality to the problem of learning task precedence graphs and automatic disassembly planning. Pomares et al. [3] also worked on an object-oriented representation of the information required for the determination of the disassembly movements. They included the information of the tools and places that allow a manipulator to grasp and do the disassembly. Li et al. [4] presented a desktop VR application for disassembly training for maintenance tasks.

This paper addresses two issues unresolved by the work described above. The first is the difficulty of gathering data required to estimate the parameters used in mathematical models of the disassembly process, specifically data related to the potential for causing damage during disassembly. The second is the effect of the residual, unavoidable uncertainty associated with that data.

3 Disassembly Sequence Model Formulation

Much of the previous literature has considered disassembly sequencing as a single objective problem. In the current research, disassembly sequencing is regarded as a multiattribute, rather than a multi-objective problem. The most commonly employed approach to exploring tradeoffs between attributes is to present the designer with a graphical depiction of the Pareto optimal frontier, that shows the set of feasible design alternatives where it is not possible to improve one attribute without adversely affecting another [28,29].

The next step is to determine what single solution on the optimal frontier represents the best outcome. The simplest approach is to define one attribute as "most important," and select the alternative that is best in that attribute. If there are more than two such alternatives, the one that is best in the second most important attribute is chosen, and so on. A more balanced approach is to identify the best c*ombination* of attributes, typically by determining the willingness to make tradeoffs among attributes by

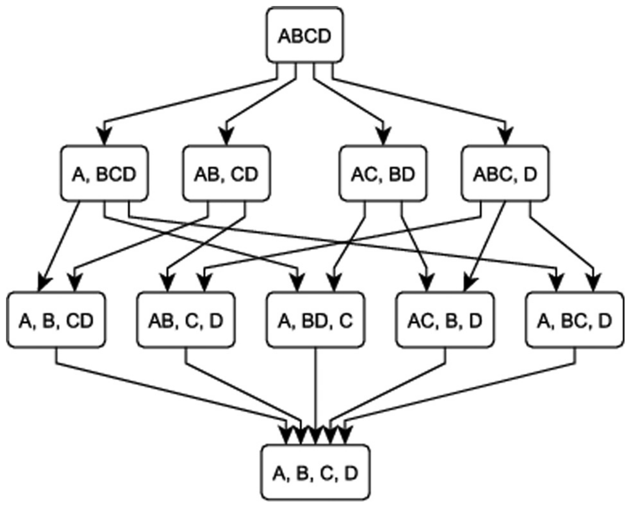

Fig. 1 Disassembly graph based on corresponding subassembly states

assigning weighting factors, which are typically interpreted to reflect relative importance. While this heuristic is better than myopically focusing on only the most important attribute and is a reasonable first attempt at determining appropriate tradeoffs, it has been demonstrated to result in choices that do not reflect the designers true preference structure [30]. Methods that employ normative multiattribute utility analysis can be used to solve this problem [31]. Using this approach, tradeoffs can be quantified with reasonable accuracy, uncertainty and its effect can be quantified, and both tradeoffs and uncertainty decisions can be fully integrated into the disassembly sequence decision making process.

The first step in determining the optimal disassembly sequence is to define the feasible disassembly transitions/alternatives. Disassembly graphs can be driven based on the information of coherence and detachability. They represent the generation of all the possible disassembly sequences. After constructing the disassembly graph, the search for reasonable sequences begins, which can be done according to heuristic criteria [2]. In the current research, the optimum disassembly sequences are generated through the application of dynamic programming with a utility value assigned to each disassembly action (arc of the graph). Figure 1 shows an example of a disassembly graph for a simple assembly with four components. Corresponding subassembly states are listed at each node of the graph. The set of disassembly choices is condensed in a single disassembly graph that is based on connective states and disassembly actions are transitions between these states.

Defining the relevant and negotiable attributes is the next step. Here, we will consider two attributes; the cost of performing each disassembly transition and the probability of incurring damage during that transition.

A dynamic programming model [32] is then used to determine the sequence of disassembly transitions that result in the optimal, or best, combination of conflicting attributes. The goal is to find a path with maximum utility considering the whole product. The basic idea of a dynamic program is to define stages and states and then use backward or forward recursion methods to determine the optimal decisions in each stage. The decision in each stage (i) is to choose the path or the optimal state(s) in the next stage ($i+1$) which results in the maximum utility. A backward recursion method can be applied to choose the optimal path at each stage. The process starts from the final node of the graph and return to the starting node.

The index set and model parameters are defined as follows:

Index set:

J: the set of all disassembly transitions (all edges in disassembly graph)
I: the set of all stages
x: first attribute
y: second attribute
i: stage i
s: state s in stage i
S: the set of all states in stage i
n: state n in stage $i+1$
N: the set of all states in stage $i+1$
j: feasible disassembly transition (action)

Parameters:

x_j: cost incurred during disassembly transition j
y_j: probability of damage incurred during disassembly transition j
k_x: scaling constant for attribute x
k_y: scaling constant for attribute y
$U_j(x)$: utility of attribute x for disassembly transition j
$U_j(y)$: utility of attribute y for disassembly transition j
$U_j(x, y)$: the two-attribute utility function for disassembly transition j
$U_j(x, y, s_i, n_{i+1})$: the utility of transition j from state s in stage i to state n in stage $i+1$
$f_i(s)$: the maximum utility from states s in stage i (nodes in stage i) to the destination node (last disassembly level)

Eq. (1) shows the dynamic programming model that maximizes the two-attribute utility

$$f_i(s) = \underset{j \in J}{\text{Max}} \{U_j(x, y, s_i, n_{i+1}) + f_{i+1}(n)\} \tag{1}$$

To find the optimal path in the disassembly graph, the two-attribute utility values associated with each arc of the disassembly graph needs to be estimated. To estimate the two-attribute utility value we need to define the single utility function for each attribute.

For engineering design formulations, considerations for defining the appropriate set of attributes, testing independence conditions, determining the form of the multiattribute utility function, and assessing its elements $U(x)$, $U(y)$ and k_x and k_y have been presented elsewhere in detail [30,31,33] and will not be repeated here. Equation (2) shows a two-attribute utility function written as a composition of two single attribute utility functions. Therefore, the two-attribute utility of disassembly transition j can be calculated as follows:

Fig. 2 A simple six piece Burr puzzle

Table 1 Desirable features of Burr puzzle for use in ICT disassembly simulation

Desirable feature	Burr puzzle characteristics
Number of components and variety of components	Six unique components
Movement limitation	Each component may move in one, two, or three equally orthogonal directions
Interlocking limitations	Assembled components are interlocked with one another providing for sequence based disassembly
Multiple disassembly sequences	Affords many deterministic disassembly sequences. The "obvious" sequence may not be optimal.
Multiple disassembly operation types	Component removal (component is removed from product assembly) and/or component reconfiguration (part is reoriented, but remains a part of the product assembly)
Partial disassembly into subassemblies	Multiple opportunities during disassembly to create subassemblies of several pieces

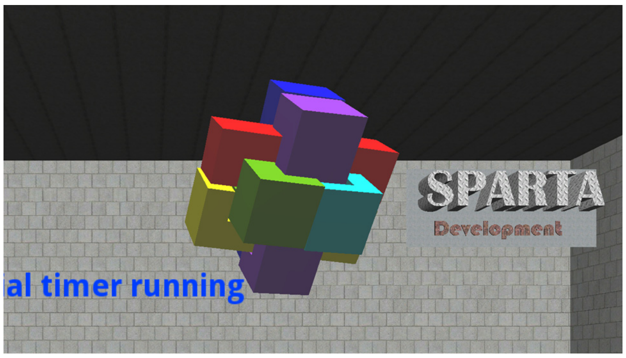

Fig. 3 An assembly view of the Burr puzzle in an ICT environment

$$U_j(x,y) = k_x U_j(x) + k_y U_j(y) + (1 - k_x - k_y) U_j(x) U_j(y) \tag{2}$$

It should be noted that the independence conditions of utility analysis are not related to the interdependency of the attributes. Although the speed of disassembly and the amount of components damage are interdependent, this does not necessarily violate the "utility independence" assumption applied in Eq. (2). In fact, the independence condition of utility analysis has nothing to do with the interdependency or independency of the attributes, but rather with preferences for attributes [31].

In the case of uncertain attribute outcomes, the utility functio-nU_j (x) or U_j (y) can be replaced by expected utility shown in Eq. (3), applying the probability density functions $f(x)$ and $f(y)$, given probabilistic independence

$$EU(x) = \int f(x)U(x)dx \tag{3}$$

4 Example: Burr Puzzle

Burr puzzles are a collection of interlocking puzzles traditionally made of wood. A simple six piece burr puzzle as shown in Fig. 2 is used as the test bed application to demonstrate the method proposed in this paper.

Because of their unique geometric properties, Burr puzzles provide an interesting assembly/disassembly test bed, for reasons listed in Table 1.

The movement of each piece is limited given the interlocking nature of the assembly configuration. Further, the movement of each piece is constrained along orthogonal axes. Users generally assume there is one "correct" assembly method, while in fact there are multiple assembly sequences. The additional sequences arise when considering the potential for disassembling the component into sub assemblies (as compared to always removing only one piece) and considering reorientation of a piece to afford removal of another piece.

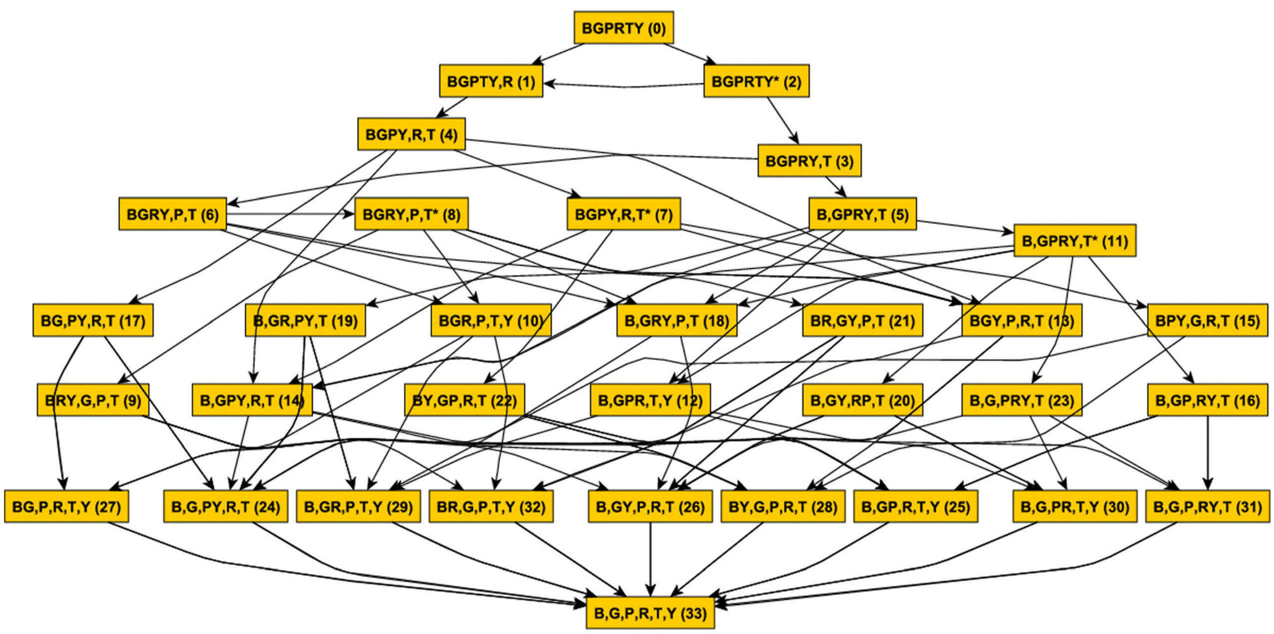

Fig. 4 Disassembly graph of the six piece Burr puzzle

Table 2 The states of each stage in the Burr puzzle disassembly graph

Stage	State(s)
1	0
2	2
3	1, 3
4	5
5	4, 11, 6
6	7, 20, 8
7	9, 10, 12, 13, 14, 15, 16, 17, 18, 19, 21, 22, 23
8	24, 25, 26, 27, 28, 29, 30, 31, 32
9	33

To develop the ICT environment, the individual puzzle pieces were modeled as 3D objects using GOOGLE SKETCHUP. For the purposes of increasing visual distinctiveness in the immersive environment, and to easily correlate parts shapes to actions, each piece was given a unique label (red, teal, blue, green, purple, and yellow) (Fig. 3).

A disassembly graph is created by manipulating the real puzzle to determine all of the possible assembly sequences (Fig. 4). The first letter of the label of a block is the identifier used for each block. A completed puzzle can be represented in the graph as "BGPRTY" (Blue, Green, Purple, etc.). The transitions between states consist of either part removal or part repositioning without removal For example, the notation BGPTY, R indicates that part R has been removed from the assembly and only BGPTY remain assembled. The notation "BGPRTY*" indicates that the *Y* part has been repositioned but not removed.

The disassembly graph of the Burr puzzle shown in Fig. 4 consists of 9 stages and 33 possible states. Table 2 summarizes the states associated with each stage. As can be seen in Fig. 4, although the number of components is small, the Burr puzzle provides a reasonable number of feasible disassembly sequences. Often in reality, as a result of precedence relationships of disassembly operation steps, the complex products with a high number of components provide far fewer number of feasible disassembly sequences than the number that would result if there were no precedence relationships. Therefore, in terms of the number of disassembly alternatives, the Burr puzzle serves as a good example of reality-complicated assembly.

The purpose of using the ICT environment is to have a user actually disassemble the part and collect the data needed for the optimization problem. There are two attributes being considered in this example: cost due to time of disassembly, and probability of damage incurred during the disassembly process. The state-of-the-art of simulating part interactions using ICT is such that absolute timing of the disassembly process is not possible. Research has not validated that task time using ICT is directly correlated with actual task time with real objects. Therefore, to provide a measure of time of disassembly, the distance of movement during a disassembly task is used as a surrogate measure. Movement along each of three orthogonal axes can be generated during an ICT disassembly task. Estimating the probability of damage during a task is also not possible to measure directly using ICT. Here, we have chosen to estimate this attribute by correlating it to the number of collisions of the 3D models during disassembly. The rationale lies in the belief that more collisions have the potential to cause more damage during a disassembly task. Although applying the distance of movement as a measure for disassembly time and applying the number of collisions as a proxy for the components damage are far from the ideal, considering the limitations of the existing simulation technologies, these proxies provide helpful information for comparing different disassembly alternatives especially at the early stage of the design in which the actual prototype of the product does not exist and in some cases is very expensive to build.

Fig. 5 Immersive virtual environment for disassembly

To facilitate disassembly modeling, the ICT used in this research includes stereo viewing, position tracking of the head and a haptic device to render collision forces to the user during the simulation (Fig. 5). This environment allows the user to select objects and manipulate them while holding the haptic device. Collision forces guide the user as to how to manipulate each object to accomplish disassembly. Sometimes disassembly operations require reorientation of the product. In this example, we assumed that the Burr puzzle is fixed in a certain position so that several operations can be conducted. Moreover, in each trial of the experiment the disassembly sequence is given; therefore, the user does not need to explore intuitive and feasible disassembly sequences. Based on the given sequence, the disassembly is carried out and the required data are recorded.

Each 3D object is modeled both as a collection of polygons (for visual rendering) and a collection of volume elements or voxels (for dynamic simulation) (Fig. 6). The size of the voxels may be specified during voxelization which is a procedure used to generate the voxelized model from the geometry model. Collisions are calculated on a voxel-to-voxel basis. When a user moves one object in contact with another, the number of voxel collisions is tallied and recorded. A collision between objects may be recorded as several thousand voxel collisions. The collision calculations are computed at one thousand times per second. Collisions are only summed when a particular part is virtually manipulated. This ensures that the collisions of two pieces which might be resting upon each other are not included in the summation.

Several studies have offered computational algorithms to determine collision free paths for physical robots and virtual agents [34–36]. The purpose of these studies is often motion planning of robots. Although disassembly is automated in some cases, in practice disassembly is labor intensive and conducted manually [37]. Therefore, the purpose of the current study is to simulate manual disassembly in the virtual environment in order to provide input to the design of the product. Using the data collected from the ICT simulation, the multiattribute utility theory can be employed to make tradeoffs between component damage (number of collisions) and disassembly time (distance of movement) as two different objectives in disassembly operations. Multiattribute utility theory is helpful in handling the tradeoffs among multiple

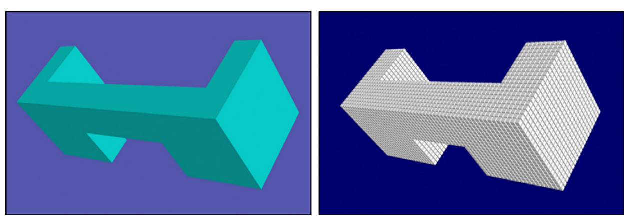

Fig. 6 (*a*) Polygonal representation and (*b*) voxel representation

Table 3 Burr Puzzle disassembly state transitions with estimated distance cost

Initial state	Disassembly operation	Resulting state	Movement (mm) X	Y	Z	Distance cost (mm)
0	−R	1	0	0	72	72
0	=R	2	0	0	24	24
1	−T	4	0	12	12	24
2	−T	3	0	12	12	24
2	−R	1	0	0	48	48
3	−B	5	12	0	12	24
3	−P	6	12	0	12	24
4	−P	13	12	0	12	24

objectives, particularly when there is uncertainty in disassembly time and probability of damage.

5 Results

This section first presents the data gathered using the ICT to simulate the disassembly process, then presents the results of incorporating that data into the decision model.

To estimate disassembly time, the distance that each part was moved by the user in the immersive environment during a given operation was measured. The user manipulated the puzzle pieces using ICT and estimated the distance a given part had to be moved in the x, y, or z direction for each step in the disassembly process. To arrive at an estimate for disassembly time for a given transition from one state to the next, the distances covered during that transition were added. For example, to transition from state 1 to state 4 requires the removal of the teal ("T") piece. The manipulation of this piece is 12 mm in the y direction and 12 mm in the z direction resulting in a total distance (cost) of 24 mm. Table 3 shows the resulting data for states 0 – state 4.

To estimate the probability of damage caused by each manipulation, the number of collisions that occurred for each transition was recorded. The burr puzzle was disassembled by an individual, using the ICT, and the voxel collisions were tabulated. The results for three trials were averaged. Figure 7 presents the average collisions per disassembly step for each of the 99 possible manipulations in the entire disassembly graph. The "x" axis is organized to show each transition as a user works through the disassembly graph performing the required transitions to move from stage to stage. The figure clearly shows that there are multiple collisions at the beginning of the disassembly process when there are more parts in the assembly, and fewer collisions toward the end of the disassembly process where only a few parts remain.

After using ICT to gather this data, the dynamic programming decision tradeoff model from Eq. (1) was used to determine the optimal disassembly sequence.

Equation (4) shows the exponential utility function reflecting risk aversion that is used for $U(x)$.

$$U(x) = a + be^{-\gamma x} \tag{4}$$

The risk aversion coefficient γ reflects the decision maker's degree of risk aversion, and the constants a and b are calculated to normalize $U(x)$ from 0 to 1, where $U(x) = 0$ when cost is the worst that the decision maker is willing to consider tolerating, and $U(x) = 1$ when cost is the best possible (least) cost.

In the Burr puzzle example, movement ranges over the interval from 0 mm to 120 mm, and $U(x)$ for the part movement as a measure of disassembly cost is

$$U(x) = -0.42 + 1.43e^{-0.01x} \tag{5}$$

Equations (6) and (7) show the linear utility function (reflecting risk neutrality toward the probability of damage) that is assumed for $U(y)$.

$$U(y) = \frac{\text{y} - \text{Worst value}}{\text{Best Value} - \text{Worst Value}} \tag{6}$$

$$U(\text{y}) = \frac{y - 3557}{10 - 3557} \tag{7}$$

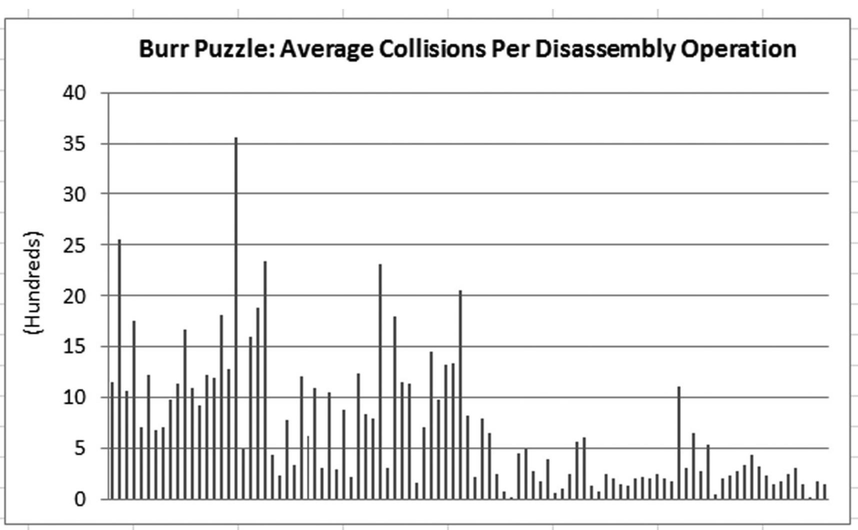

Fig. 7 Average collision data for each transition in the disassembly tree

Table 4 Sample cost data and utility function values

s	j	n	Disassembly movement (x)/mm	Damage (y)/count	$U_j(x)$	$U_j(y)$	$U_j(x, y)$
0	−R	1	72	1155	0.28	0.68	0.49
	=R	2	24	2560	0.70	0.28	0.40
1	−T	4	24	1066	0.70	0.70	0.68
2	−T	3	24	1751	0.70	0.51	0.55
	−R	1	48	710	0.46	0.80	0.65
3	−B	5	24	1226	0.70	0.66	0.65
	−P	6	24	669	0.70	0.81	0.75
4	−P	13	24	710	0.70	0.80	0.74
	−B	14	24	972	0.70	0.73	0.70
	S	17	48	1139	0.46	0.68	0.57
	=B	7	12	1666	0.85	0.53	0.62
5	−R	14	48	1087	0.46	0.70	0.58
	−P	18	24	921	0.70	0.74	0.70
	−Y	12	5	1215	0.94	0.66	0.74
	S	19	48	1196	0.46	0.67	0.56
	=R	11	60	1804	0.36	0.49	0.42
6	−R	13	48	1275	0.46	0.64	0.55
	−B	18	24	3557	0.70	0.00	0.22
	−Y	10	12	482	0.85	0.87	0.85
	=R	8	36	1599	0.58	0.55	0.53
.	.	.					
.	.	.					
.	.	.					
31	−R	33	12	144	0.85	0.96	0.91
	−Y	33	12	18	0.85	1.00	0.93
32	−B	33	12	176	0.85	0.95	0.90
	−R	33	12	141	0.85	0.96	0.91
33		Final node					

Where, the worst value for the number of collisions for Burr puzzle example is 3557 and the best value is 10.

Table 4 presents a partial set of data generated from the ICT simulation and resulting elements of the dynamic programming model. Column 1 indicates the state s, column 2 indicates transitions j where "−R" indicates removal of the red part and "=R" indicates repositioning of the red part, column 3 indicates the resulting state n, columns 4 and 5 show the resulting distance x and number of collisions y, columns 6 and 7 show the resulting single attribute utilities $U_j(x)$ and $U_j(y)$ for cost and probability of damage, respectively. The last column shows the $U(x, y)$ resulting from employing scaling constant values k_x and k_y, which reflect the decision maker's willingness to make tradeoffs between cost and the probability of damage. Lottery method can be applied to assess the scaling constants [38]. For the purpose of this example, scaling constants are assumed to be approximately $k_x = 0.30$ and $k_y = 0.60$.

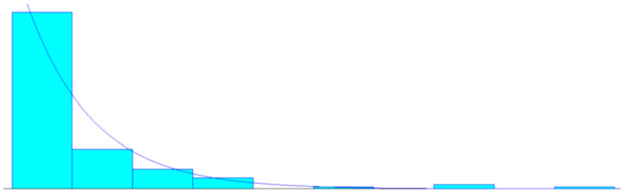

Fig. 9 The statistical distribution of the number of collisions in transition 2-3

Figure 8 shows the results of using Eq. (1) to solve the dynamic programming model, giving the optimal route from node 0 to node 33. The result indicates that the sequence from $0 \rightarrow 2 \rightarrow 1 \rightarrow 4 \rightarrow 7 \rightarrow 14 \rightarrow 26 \rightarrow 33$ is the optimal disassembly sequence for the Burr puzzle example.

The disassembly sequence obtained here was based on single data points for distance of movement and number of collisions. To consider the uncertainties as a result of operator's dexterity and manipulability, more data can be generated by conducting each disassembly transition more than one trial and the statistical distributions that best fit to data can be identified. Finally, the utility functions can be replaced by expected utility function applying the statistical distributions of data.

As an example, consider the operation in which component T is removed from the whole assembly. The arc connecting node 2 to 3 in Fig. 8 illustrates this disassembly operation. Suppose that the disassembly movement follows a uniform distribution U(20, 28). In order to identify the uncertainty in the number of collisions, the disassembly operation was conducted 120 times and the number of collisions was recorded. Figure 9 illustrates the distribution fitted to the data points. Applying these distributions and

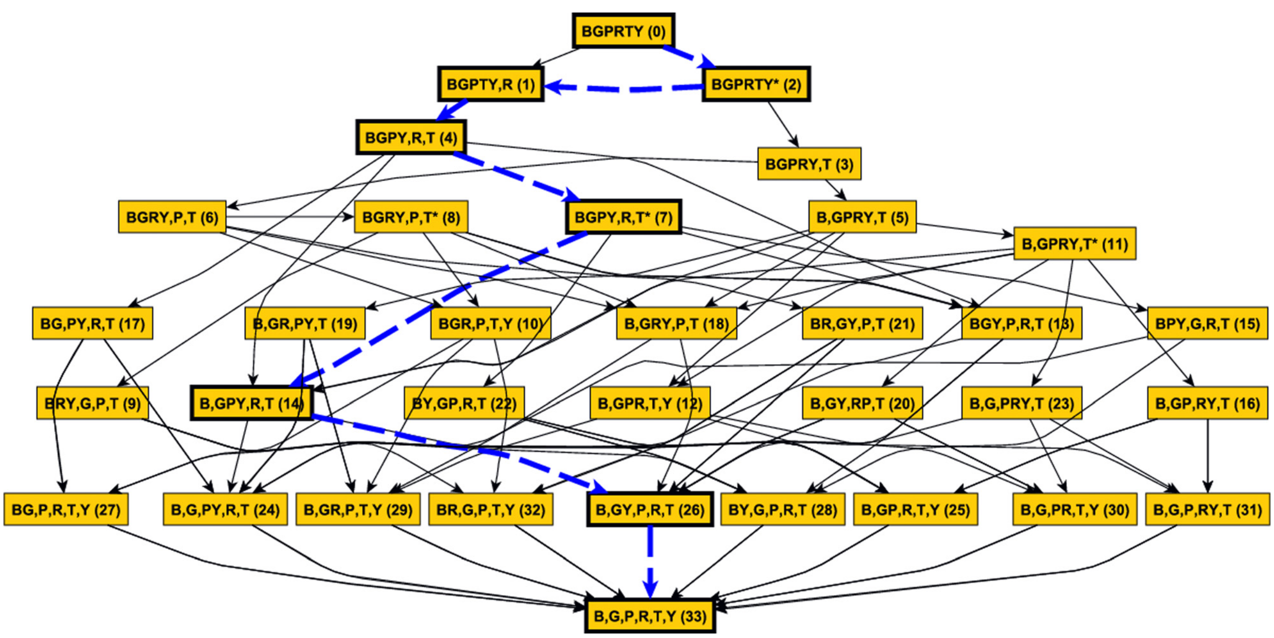

Fig. 8 Disassembly graph of the six piece Burr puzzle including the optimal disassembly route

Table 5 Utility function values for disassembly transition 2–3

Disassembly transition	Disassembly movement (x)/mm	Damage (y)/count	$EU_j(x)$	$EU_j(y)$	$EU_j(x, y)$
Node 2-3	U(20, 28)	$221 + \mathrm{EXPO}(1.97 \times 10^{+003})$	0.50	0.92	0.73

Eqs. (3), (5), and (7), the expected utilities for each individual attribute and finally the overall utility of the disassembly transition were calculated (Table 5).

6 Discussion and Future Work

This paper has presented a framework for investigating tradeoffs under uncertainty using immersive computing technology. There are two difficult aspects of uncertainty that the approach presented here addresses. The first is the difficulty in gathering data required to estimate the values of the parameters used in mathematical models of the disassembly process, in this case the time (and cost) of a large number of possible disassembly sequence steps, and the probability of damage caused while carrying out those steps. The second difficulty is that even after the data is gathered, unavoidable uncertainty remains, and the designer must determine its effect on the relative desirability of a very large number of possible design alternatives, in this case disassembly sequence steps. This paper presented a method for employing ICT to carry out a virtual experiment in order to simulate a large number of disassembly process steps, and from those simulations better estimate the cost and probability of damage associated with each possible step. Then, mathematical models (dynamic programming and multiattribute utility analysis) were employed to determine the disassembly sequence that resulted in the optimal combination of cost and probability of damage.

The ICT demonstrated an effective method to gather data on human interaction with the product that can be used to improve the decision making process. In the proposed scenario, the user manipulates the virtual parts to estimate values for potential damage that might occur during disassembly. This data is subsequently used as input to the dynamic programming decision model used to determine the optimal disassembly process. Without the ability to manipulate real parts, the designer has to rely on past experience to anticipate the extent of damage during each part removal process. ICT provides a computer generated environment that supports user manipulation of virtual CAD models, thus allowing this data to be generated prior to manufacture of actual products. Decisions about the design of the product that are affected by disassembly operations can be made prior to final product design.

There are numerous opportunities for future work. The development of a more comprehensive model for estimating component damage (from haptic interaction) would increase data reliability. In addition, most products contain various types of fasteners such as screws, rivets, and snaps. Inclusion of fasteners would require interactive simulation of deformable surfaces of the virtual models and manipulation of tools to aid in disassembly. Interactive simulation of deformable surfaces and the use of tools are common features of virtual surgery applications and could readily be implemented in this work. Finally, it would be worthwhile to investigate how ICT can be employed to overcome the systematic biases that might be embedded in cognitive heuristics that designers use to estimate the costs and damage resulting from various product design and disassembly alternatives.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant Nos. CMMI-1100177 and CMMI-1068926. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] Giudice, F., and Kassem, M., 2009, "End-of-Life Impact Reduction Through Analysis and Redistribution of Disassembly Depth: A Case Study in Electronic Device Redesign," Comput. Ind. Eng., **57**(3), pp. 677–690.

[2] Lambert, A. J. D., 2003, "Disassembly Sequencing: A Survey," Int. J. Prod. Res., **41**(16), pp. 3721–3759.

[3] Pomares, J., Puente, S. T., Torres, F., Candelas, F. A., and Gil, P., 2004, "Virtual Disassembly of Products Based on Geometric Models," Comput. Ind., **55**(1), pp. 1–14.

[4] Li, J. R., Khoo, L. P., and Tor, S. B., 2003, "Desktop Virtual Reality for Maintenance Training: an Object Oriented Prototype System (V-REALISM)," Comput. Ind., **52**(2), pp. 109–125.

[5] Hula, A., Jalali, K., Hamza, K., Skerlos, S. J., and Saitou, K., 2003, "Multi-Criteria Decision-Making for Optimization of Product Disassembly under Multiple Situations," Environ. Sci. Technol., **37**(23), pp. 5303–5313.

[6] McGovern, S. M., and Gupta, S. M., 2006, "Ant Colony Optimization for Disassembly Sequencing With Multiple Objectives," Int. J. Adv. Manuf. Technol., **30**(5-6), pp. 481–496.

[7] Lee, D.-H., Xirouchakis, P., and Zust, R., 2002, "Disassembly Scheduling With Capacity Constraints," CIRP Ann., **51**(1), pp. 387–390.

[8] Kang, J.-G., and Xirouchakis, P., 2006, "Disassembly Sequencing for Maintenance: A Survey," Proc. Inst. Mech. Eng., Part B (J. Eng. Manuf.), **220**(B10), pp. 1697–1716.

[9] Gonzalez-Torre, B., and Adenso-Diaz, B., 2003, "Optimizing Decision Making at the End of Life of a Product," Proc. SPIE **5262**, pp. 40–50.

[10] Kara, S., Pornprasitpol, P., and Kaebernick, H., 2005, "A Selective Disassembly Methodology for End-of-Life Products," Assem. Autom., **25**(2), pp. 124–134.

[11] Behdad, S., Kwak, M., Kim, H., and Thurston, D., 2010, "Simultaneous Selective Disassembly and End-of-Life Decision Making for Multiple Products That Share Disassembly Operations," ASME J. Mech. Des., **132**(4), p. 041002.

[12] Behdad, S., and Thurston, D., 2012, "Disassembly and Reassembly Sequence Planning Tradeoffs Under Uncertainty for Product Maintenance," ASME J. Mech. Des., **134**(4), p. 040201.

[13] Jayaram, S., Jayaram, U., Kim, Y. J., DeChenne, C., Lyons, K. W., Palmer, C., and Mitsui, T., 2007, "Industry Case Studies in the Use of Immersive Virtual Assembly," Virtual Reality, **11**(4), pp. 217–228.

[14] Seth, A., Vance, J. M., and Oliver, J. H., 2011, "Virtual Reality for Assembly Methods Prototyping: A Review," Virtual Reality, **15**(1), pp. 5–20.

[15] Jayaram, S., Connacher, H. I., and Lyons, K. W., 1997, "Virtual Assembly Using Virtual Reality Techniques," CAD, **29**(8), pp. 575–584.

[16] Seth, A., Su, H.-J., and Vance, J. M., 2006, "Sharp: A System for Haptic Assembly Realistic Prototyping BT," 2006 ASME International Design Engineering Technical Conferences and Computers and Information In Engineering Conference, American Society of Mechanical Engineers, Department of Mechanical Engineering, Virtual Reality Applications Center, Iowa State University, Ames, IA, Paper No. DETC2006.

[17] Lin, M. C., and Manocha, D., 1995, "Fast Interference Detection Between Geometric Models," Visual Comput., **11**(10), pp. 542–561.

[18] Jimenez, P., Thomas, F., and Torras, C., 2001, "3D Collision Detection: A Survey," Comput. Graphics, **25**(2), pp. 269–285.

[19] Borro, D., Hernantes, J., Garcia-Alonso, A., and Matey, L., 2005, "Collision Problem: Characteristics for a Taxonomy," Proceedings. Ninth International Conference on Information Visualisation, IEEE Comput. Soc, Ceit, Manuel de Lardizabal, San Sebastian, Spain, July 6–8, BN–0 7695 2397 8, pp. 410–415.

[20] Kim, C. E., and Vance, J. M., 2004, "Collision Detection and Part Interaction Modeling to Facilitate Immersive Virtual Assembly Methods," ASME J. Comput. Inf. Sci. Eng., **4**(2), pp. 83–90.

[21] Faas, D., and Vance, J. M., 2010, "Assessment of Pointshell Shrinking and Feature Size on Virtual Manual Assembly," ASME 2010 World Conference on Innovative Virtual Reality, American Society of Mechanical Engineers, Department of Mechanical Engineering, Virtual Reality Applications Center, Iowa State University, Ames, IA, May 12–14, WINVR 2010, pp. 211–218.

[22] Seth, A., Vance, J. M., and Oliver, J. H., 2010, "Combining Dynamic Modeling With Geometric Constraint Management To Support Low Clearance Virtual Manual Assembly," ASME J. Mech. Des., **132**(8), p. 081002.

[23] Dong, J., and Arndt, G., 2003, "A Review of Current Research on Disassembly Sequence Generation and Computer Aided Design for Disassembly," Proc. Inst. Mech. Eng., Part B (J. Eng. Manuf.), **217**(B3), pp. 299–312.

[24] Ritchie, J., Simmons, J., Dewar, R., and Carpenter, I., 1999, "A Methodology for Eliciting Expert Knowledge in Virtual Engineering Environments," Proceedings of Portland International Conference on Management of Engineering and Technology, Portland Int. Conf. Manage. Eng. Technol. PICMET, Dept. of Mech. Chem. Eng., Heriot-Watt Univ., Edinburgh, UK, July 25–29, Vol. 1, BN–1 890843 02 4, p. 202.
[25] Dewar, R. G., Carpenter, I. D., Ritchie, J. M., and Simmons, J. E. L., 1997, "Assembly Planning in a Virtual Environment," Innovation in Technology Management. The Key to Global Leadership. PICMET '97, IEEE, Dept. of Mech. Chem. Eng., Heriot-Watt Univ., Edinburgh, UK, July 27–31, BN–0 7803 3574 0, pp. 664–667.
[26] Bullinger, H. J., Richter, M., and Seidel, K.-A., 2000, "Virtual Assembly Planning," Hum. Factors Ergon. Manuf., **10**(3), pp. 331–341.
[27] Aleotti, J., and Caselli, S., 2011, "Physics-Based Virtual Reality for Task Learning and Intelligent Disassembly Planning," Virtual Reality, **15**(1), pp. 41–54.
[28] Mattson, C. A., and Messac, A., 2005, "Pareto Frontier Based Concept Selection Under Uncertainty, With Visualization," Optim. Eng., **6**(1), pp. 85–115.
[29] Sengupta, M., and Styblinski, M. A., 1997, "Visualization of Trade-Offs in Optimization of Integrated Circuits With Multiple Objectives," Proceedings of 1997 IEEE International Symposium on Circuits and Systems. Circuits and Systems in the Information Age ISCAS '97, IEEE, Dept. of Electr. Eng., Texas AM Univ., College Station, TX, June 9–12, BN–0 7803 3583 X, pp. 1640–1643.
[30] Thurston, D. L., 1991, "A Formal Method for Subjective Design Evaluation With Multiple Attributes," Res. Eng. Des., **3**(2), pp. 105–122.
[31] Thurston, D. L., 2001, "Real and Misconceived Limitations to Decision Based Design With Utility Analysis," ASME J. Mech. Des., **123**(2), pp. 176–182.
[32] Bellman, R. E., 1957, *Dynamic Programming*, Princeton University Press, Princeton, NJ.
[33] Tian, Y. Q., Thurston, D. L., and Carnahan, J. V., 1994, "Incorporating End-User's Attitudes Towards Uncertainty Into an Expert System," ASME J. Mech. Des., **116**(2), pp. 493–500.
[34] Pan, J., Zhang, L., and Manocha, D., 2012, "Collision-Free and Smooth Trajectory Computation in Cluttered Environments," Int. J. Rob. Res., **31**(10), pp. 1155–1175.
[35] Lauterbach, C., Mo, Q., and Manocha, D., 2010, "gProximity: Hierarchical GPU-Based Operations for Collision and Distance Queries," Comput. Graph. Forum, **29**(2), pp. 419–428.
[36] Zhang, L., Huang, X., Kim, Y. J., and Manocha, D., 2008, "D-Plan: Efficient Collision-Free Path Computation for Part Removal and Disassembly," Comput.-Aided Des. Appl., **5**(6), pp. 774–786.
[37] Tang, Y., Zhou, M., and Gao, M., 2006, "Fuzzy-Petri-Net-Based Disassembly Planning Considering Human Factors," IEEE Trans. Syst., Man Cybern., Part A Syst. Humans, **36**(4), pp. 718–726.
[38] Thurston, D., 2006, "Multi-Attribute Utility Analysis of Conflicting Preferences," *Decision Making in Engineering Design*, K. Lewis, W. Chen, and L. C. Schmidt, eds., ASME Press, New York.